Remarks

Applicants respectfully request entry of the above amendments and consideration of the application, as amended. After entry of the amendments, claims 1 and 4-71 are pending. Claims 2 and 3 have been rewritten as claims 25 and 50, respectively. Support for the new claims can be found throughout the specification (e.g., pages 35-40), and therefore, no new matter has been added.

Additionally, paragraph 3 on page 46 has been amended to correct a grammatical error.

In addition, paragraph 1 on page 64 has been amended to correct two typographical errors. The first is to indicate the correct number of iterations for the equation. As shown in the examples, if there are 3 equidistant servers, then there are 3 iterations. Since k begins at 0, it should state: "or k=0 to the number of servers-1," as supported by the examples in the specification. The second correction is to balance the parenthesis. Support for the amendments can be found throughout the specification (e.g., pages 64-66), and therefore, no new matter has been added.

In accordance with 37 C.F.R. 1.121, a version with markings to show changes made is provided on one or more pages separate from the amendment. These pages are appended at the end of the Amendment.

Should the Examiner have any questions regarding this application, please call applicants' attorney at the below listed number.

Respectfully submitted,

Blanche E. Schiller

Blanche E. Schiller
Reg. No. 35,670

Dated: April 12, 2002

HESLIN ROTHENBERG FARLEY & MESITI, P.C.
5 Columbia Circle
Albany, New York 12203
Telephone: (518) 452-5600
Facsimile: (518) 452-5579

Version with markings to show changes made

In the Specification:

On page 1, paragraphs 2-5, lines 10-24 and page 2, paragraphs 1-3, lines 1-11 have been amended, as follows:

"Method, System And Program Products For Managing A Clustered Computing Environment," Novaes et al., (Docket No. POU9-2000-0004-US1), Serial No. [_____] 09/583,677, filed [herewith] May 31, 2000;

"Method, System And Program Products For Providing Clusters Of A Computing Environment," Novaes et al., (Docket No. POU9-2000-0021-US1), Serial No. [_____] 09/583,686, filed [herewith] May 31, 2000;

"Method, System And Program Products For Defining Nodes To A Cluster," Novaes et al., (Docket No. POU9-2000-0011-US1), Serial No. [_____] 09/583,582, filed [herewith] May 31, 2000;

"Method, System And Program Products For Ordering Lists Of Service Addresses To Provide Load Balancing Of A Clustered Environment," Novaes et al., (Docket No. POU9-2000-0010-US1), Serial No. [_____] 09/584,638, filed [herewith] May 31, 2000;

"Method, System And Program Products For Controlling System Traffic Of A Clustered Computing Environment," Novaes

et al., (Docket No. POU9-2000-0008-US1), Serial No. [_____] 09/583,849, filed [herewith] May 31, 2000;

"Method, System And Program Products For Automatically Configuring Clusters Of A Computing Environment," Novaes et al., (Docket No. POU9-2000-0005-US1), Serial No. [_____] 09/584,528, filed [herewith] May 31, 2000; and

"Method, System And Program Products For Managing Cluster Configurations," Novaes et al., (Docket No. POU9-2000-0096-US1), Serial No. [_____] 09/583,693, filed [herewith] May 31, 2000.

Paragraph 3 on page 46, lines 21-26 and lines 1-2 on page 47 has been amended as follows:

Thereafter, the resource controllers update the System Registry (Step 5, FIG. 15) with the configuration for the resources (e.g., hardware) that they control, and [notifies] notify the Resource Manager on the new node (Step 6, FIG. 15) that the update is complete. The Resource manager process then notifies the DCM (Step 8, FIG. 15), when it receives the completion status for this operation for the resource controllers that are registered with it.

Paragraph 1 on page 64, lines 1-8 has been amended as follows:

Next, the mapping index for one of the equidistant servers is calculated using a predefined equation, STEP

2408. In particular, for k=0 to the number of equidistant servers-1, the mapping index is equal to the [(node_number)mod(number_of_equidistant_servers)+k) mod(number_of_equidistant_servers)], where mod refers to the module operation defined as the integer remainder of a division operation.

In the Claims:

Claims 2 and 3 have been canceled.

New claims 4-71 have been added.

Delete paragraph 1 on page 64, lines 1-8 and replace with:

Next, the mapping index for one of the equidistant servers is calculated using a predefined equation, STEP 2408. In particular, for k=0 to the number of equidistant servers-1, the mapping index is equal to the [((node_number)mod(number_of_equidistant_servers)+k) mod(number_of_equidistant_servers)], where mod refers to the module operation defined as the integer remainder of a division operation.

In the Claims:

Please cancel claims 2 and 3, without prejudice.

Please add new claims 4-71. All claims are reproduced herein for the Examiner's convenience.

1. (UNCHANGED) A method of managing identifiers of components of a distributed computing environment, said method comprising:

providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of said distributed computing environment to a cluster of said distributed computing environment; and

automatically managing by the cluster said unique identifier.

2. (CANCELED)

3. (CANCELED)

4. (NEW) The method of claim 1, wherein said automatically managing is performed by a distributed configuration manager of the cluster.

B'

A4 Cont

5. (NEW) The method of claim 1, wherein said automatically managing comprises updating a local identifier and a global identifier to reflect a change of the unique identifier.

6. (NEW) The method of claim 1, wherein said automatically managing comprises:

determining, in response to an event, whether said unique identifier is consistent with at least one of a local identifier and a global identifier; and

performing an action in response to the determining.

7. (NEW) The method of claim 6, wherein said event comprises a joining of said operating system instance to said cluster.

8. (NEW) The method of claim 6, wherein said determining comprises:

comparing said unique identifier to said local identifier; and

comparing said local identifier to said global identifier.

B1
A4 Cont

9. (NEW) The method of claim 8, wherein said performing an action comprises allowing the event to proceed, in response to the unique identifier, the local identifier and the global identifier being consistent.

10. (NEW) The method of claim 8, wherein said performing an action comprises updating the local identifier to reflect that the operating system instance has been deleted from the cluster, in response to the unique identifier being equal to the local identifier, and the local identifier being unequal to the global identifier.

11. (NEW) The method of claim 8, wherein said performing an action comprises updating the local identifier and the global identifier, in response to the unique identifier being unequal to the local identifier, and the local identifier being equal to the global identifier.

12. (NEW) The method of claim 8, wherein said performing an action comprises updating the local identifier to reflect that the operating system instance has been

deleted from the cluster, in response to the unique identifier being unequal to the local identifier, and the local identifier being unequal to the global identifier.

13. (NEW) The method of claim 6, wherein said determining comprises comparing the local identifier to the global identifier.

14. (NEW) The method of claim 13, wherein said performing an action comprises updating the local identifier, in response to the local identifier being unequal to the global identifier.

15. (NEW) The method of claim 14, wherein said local identifier is updated to reflect that the operating system instance has been deleted from the cluster.

16. (NEW) The method of claim 6, further comprising storing said unique identifier in local storage to provide the local identifier.

17. (NEW) The method of claim 16, wherein said storing is performed in response to said operating system instance being defined to the cluster.

18. (NEW) The method of claim 6, further comprising storing said unique identifier in global storage to provide the global identifier.

19. (NEW) The method of claim 18, wherein said storing is performed in response to said operating system instance being defined to the cluster.

20. (NEW) The method of claim 1, wherein said automatically managing comprises:

determining whether said unique identifier is in agreement with other data; and

taking action in response to the determining.

21. (NEW) The method of claim 20, wherein said other data comprises at least one of a local copy of the unique identifier and a global copy of the unique identifier.

22. (NEW) A method of managing identifiers of components of a distributed computing environment, said method comprising:

providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of the distributed computing environment to a cluster of the distributed computing environment;

storing, by the cluster, the unique identifier in local storage and global storage, providing a local identifier and a global identifier;

determining, in response to a cluster event, whether the unique, local and global identifiers are in agreement; and

performing an action in response to the determining indicating one or more of the identifiers are not in agreement.

B1

A24 cont

23. (NEW) The method of claim 22, wherein said cluster event comprises a join of the operating system instance to the cluster.

24. (NEW) A method of managing identifiers of components of a distributed computing environment, said method comprising:

identifying a component of the distributed computing environment by a unique identifier, a local identifier and a global identifier; and

automatically updating, by a cluster of the distributed computing environment, one or more of the unique identifier, the local identifier and the global identifier, to provide consistency among the unique identifier, the local identifier and the global identifier, in response to a cluster event.

25. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

means for providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of said distributed computing environment to a cluster of said distributed computing environment; and

means for automatically managing by the cluster said unique identifier.

A4 B' Cont

26. (NEW) The system of claim 25, wherein said means for automatically managing comprises a distributed configuration manager of the cluster.

27. (NEW) The system of claim 25, wherein said means for automatically managing comprises means for updating a local identifier and a global identifier to reflect a change of the unique identifier.

28. (NEW) The system of claim 25, wherein said means for automatically managing comprises:

means for determining, in response to an event, whether said unique identifier is consistent with at least one of a local identifier and a global identifier; and

means for performing an action in response to the determining.

29. (NEW) The system of claim 28, wherein said event comprises a joining of said operating system instance to said cluster.

30. (NEW) The system of claim 28, wherein said means for determining comprises:

means for comparing said unique identifier to said local identifier; and

means for comparing said local identifier to said global identifier.

31. (NEW) The system of claim 30, wherein said means for performing an action comprises means for allowing the event to proceed, in response to the unique identifier, the local identifier and the global identifier being consistent.

32. (NEW) The system of claim 30, wherein said means for performing an action comprises means for updating the local identifier to reflect that the operating system instance has been deleted from the cluster, in response to the unique identifier being equal to the local identifier, and the local identifier being unequal to the global identifier.

33. (NEW) The system of claim 30, wherein said means for performing an action comprises means for updating the local identifier and the global identifier, in response to the unique identifier being unequal to the local identifier, and the local identifier being equal to the global identifier.

34. (NEW) The system of claim 30, wherein said means for performing an action comprises means for updating the local identifier to reflect that the operating system instance has been deleted from the cluster, in response to the unique identifier being unequal to the local identifier, and the local identifier being unequal to the global identifier.

35. (NEW) The system of claim 28, wherein said means for determining comprises means for comparing the local identifier to the global identifier.

36. (NEW) The system of claim 35, wherein said means for performing an action comprises means for updating the local identifier, in response to the local identifier being unequal to the global identifier.

37. (NEW) The system of claim 36, wherein said local identifier is updated to reflect that the operating system instance has been deleted from the cluster.

38. (NEW) The system of claim 28, further comprising means for storing said unique identifier in local storage to provide the local identifier.

39. (NEW) The system of claim 38, wherein the storing is performed in response to said operating system instance being defined to the cluster.

40. (NEW) The system of claim 28, further comprising means for storing said unique identifier in global storage to provide the global identifier.

41. (NEW) The system of claim 40, wherein the storing is performed in response to said operating system instance being defined to the cluster.

42. (NEW) The system of claim 25, wherein said means for automatically managing comprises:

means for determining whether said unique identifier is in agreement with other data; and

means for taking action in response to the determining.

43. (NEW) The system of claim 42, wherein said other data comprises at least one of a local copy of the unique identifier and a global copy of the unique identifier.

44. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

means for providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of the distributed computing environment to a cluster of the distributed computing environment;

means for storing, by the cluster, the unique identifier in local storage and global storage, providing a local identifier and a global identifier;

means for determining, in response to a cluster event, whether the unique, local and global identifiers are in agreement; and

means for performing an action in response to the determining indicating one or more of the identifiers are not in agreement.

45. (NEW) The system of claim 44, wherein said cluster event comprises a join of the operating system instance to the cluster.

46. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

means for identifying a component of the distributed computing environment by a unique identifier, a local identifier and a global identifier; and

means for automatically updating, by a cluster of the distributed computing environment, one or more of the unique identifier, the local identifier and the global identifier, to provide consistency among the unique identifier, the local identifier and the global identifier, in response to a cluster event.

47. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

an operating system instance of said distributed computing environment to provide a unique identifier of a component of said distributed computing environment to a cluster of said distributed computing environment; and

a distributed configuration manager of the cluster to manage said unique identifier.

48. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

an operating system instance of said distributed computing environment to provide a unique identifier of a component of the distributed computing environment to a cluster of the distributed computing environment;

local storage and global storage of the distributed computing environment to store the unique identifier, providing a local identifier and a global identifier;

a distributed configuration manager of the cluster to determine, in response to a cluster event, whether the unique, local and global identifiers are in agreement, and to perform an action in response to the determining indicating one or more of the identifiers are not in agreement.

49. (NEW) A system of managing identifiers of components of a distributed computing environment, said system comprising:

a component of the distributed computing environment identified by a unique identifier, a local identifier and a global identifier; and

a cluster of the distributed computing environment to automatically update one or more of the unique identifier, the local identifier and the global identifier, to provide consistency among the unique identifier, the local identifier and the global identifier, in response to a cluster event.

50. (NEW) At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing identifiers of components of a distributed computing environment, said method comprising:

providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of said distributed computing environment to a cluster of said distributed computing environment; and

automatically managing by the cluster said unique identifier.

51. (NEW) The at least one program storage device of claim 50, wherein said automatically managing is performed by a distributed configuration manager of the cluster.

52. (NEW) The at least one program storage device of claim 50, wherein said automatically managing comprises updating a local identifier and a global identifier to reflect a change of the unique identifier.

53. (NEW) The at least one program storage device of claim 50, wherein said automatically managing comprises:

determining, in response to an event, whether said unique identifier is consistent with at least one of a local identifier and a global identifier; and

performing an action in response to the determining.

54. (NEW) The at least one program storage device of claim 53, wherein said event comprises a joining of said operating system instance to said cluster.

55. (NEW) The at least one program storage device of claim 53, wherein said determining comprises:

comparing said unique identifier to said local identifier; and

comparing said local identifier to said global identifier.

56. (NEW) The at least one program storage device of claim 55, wherein said performing an action comprises allowing the event to proceed, in response to the unique identifier, the local identifier and the global identifier being consistent.

57. (NEW) The at least one program storage device of claim 55, wherein said performing an action comprises updating the local identifier to reflect that the operating system instance has been deleted from the cluster, in response to the unique identifier being equal to the local identifier, and the local identifier being unequal to the global identifier.

58. (NEW) The at least one program storage device of claim 55, wherein said performing an action comprises updating the local identifier and the global identifier, in response to the unique identifier being unequal to the local identifier, and the local identifier being equal to the global identifier.

59. (NEW) The at least one program storage device of claim 55, wherein said performing an action comprises updating the local identifier to reflect that the operating system instance has been deleted from the cluster, in response to the unique identifier being unequal to the local

identifier, and the local identifier being unequal to the global identifier.

60. (NEW) The at least one program storage device of claim 53, wherein said determining comprises comparing the local identifier to the global identifier.

61. (NEW) The at least one program storage device of claim 60, wherein said performing an action comprises updating the local identifier, in response to the local identifier being unequal to the global identifier.

62. (NEW) The at least one program storage device of claim 61, wherein said local identifier is updated to reflect that the operating system instance has been deleted from the cluster.

63. (NEW) The at least one program storage device of claim 53, wherein said method further comprises storing said unique identifier in local storage to provide the local identifier.

64. (NEW) The at least one program storage device of claim 63, wherein said storing is performed in response to said operating system instance being defined to the cluster.

65. (NEW) The at least one program storage device of claim 53, wherein said method further comprises storing said unique identifier in global storage to provide the global identifier.

66. (NEW) The at least one program storage device of claim 65, wherein said storing is performed in response to said operating system instance being defined to the cluster.

67. (NEW) The at least one program storage device of claim 50, wherein said automatically managing comprises:

determining whether said unique identifier is in agreement with other data; and

taking action in response to the determining.

AA B1 Cm.1

68. (NEW) The at least one program storage device of claim 67, wherein said other data comprises at least one of a local copy of the unique identifier and a global copy of the unique identifier.

69. (NEW) At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing identifiers of components of a distributed computing environment, said method comprising:

providing, by an operating system instance of said distributed computing environment, a unique identifier of a component of the distributed computing environment to a cluster of the distributed computing environment;

storing, by the cluster, the unique identifier in local storage and global storage, providing a local identifier and a global identifier;

B1
Aft Cncld.

determining, in response to a cluster event, whether the unique, local and global identifiers are in agreement; and

performing an action in response to the determining indicating one or more of the identifiers are not in agreement.

70. (NEW) The at least one program storage device of claim 69, wherein said cluster event comprises a join of the operating system instance to the cluster.

71. (NEW) At least one program storage device readable by a machine tangibly embodying at least one program of instructions executable by the machine to perform a method of managing identifiers of components of a distributed computing environment, said method comprising:

identifying a component of the distributed computing environment by a unique identifier, a local identifier and a global identifier; and

automatically updating, by a cluster of the distributed computing environment, one or more of the unique identifier, the local identifier and the global identifier, to provide consistency among the unique identifier, the local identifier and the global identifier, in response to a cluster event.